NONLINEAR PROGRAMMING

K. Schittkowski and **Ch. Zillober**
University of Bayreuth, Germany

Keywords: nonlinear programming, optimization, sequential quadratic programming, sequential linear programming, sequential convex programming, penalty method, barrier method, reduced gradient method, interior point method, optimality, Karush-Kuhn-Tucker condition, convergence, large scale optimization

Contents

1. Introduction
2. Optimality Conditions
2.1 Convexity and Constraint Qualification
2.2. Karush-Kuhn-Tucker Conditions
3. Optimization Algorithms
3.1. Quasi-Newton Methods for Unconstrained Optimization
3.2 Penalty and Barrier Methods
3.3 Augmented Lagrangian Methods
3.4 Interior Point Methods
3.5 Sequential Linear Programming
3.6 Sequential Quadratic Programming
3.7 Generalized Reduced Gradient Methods
3.8 Sequential Convex Programming
4. Large Scale Optimization
4.1 Limited Memory Quasi-Newton Updates
4.2 Sequential Linear Programming
4.3 Interior Point Methods
4.4 Sequential Quadratic Programming
4.5 Sequential Convex Programming
Glossary
Bibliography
Biographical Sketches

Summary

Nonlinear programming is a direct extension of linear programming, when we replace linear model functions by nonlinear ones. Numerical algorithms and computer programs are widely applicable and commercially available as *black box* software. However, to understand how optimization methods work; how corresponding programs are organized; how the results are to be interpreted; and, last but not least, what are the limitations of the powerful mathematical technology; it is necessary to understand at least the basic terminology. Thus, we present a brief introduction to optimization theory; in particular, we introduce optimality criteria for smooth problems. These conditions are extremely important to understanding how mathematical algorithms work. The most popular classes of constrained nonlinear programming algorithms are introduced, i.e., penalty-barrier, interior point, augmented Lagrangian, sequential quadratic

programming, sequential linear programming, generalized reduced gradient, and sequential convex programming methods. Common features and methodological differences are outlined. In particular, we discuss extensions of these methods for solving large-scale nonlinear programming problems.

1. Introduction

Whenever a mathematical model is available to simulate a *real-life* application, a straightforward technique is to apply mathematical optimization algorithms for minimizing a so-called cost function, subject to constraints. A typical example is the minimization of the weight of a mechanical structure under certain loads and constraints for admissible stresses, displacements, or dynamic responses. Highly complex industrial and academic design problems are solved by using nonlinear programming algorithms, without there being any chance of getting equally qualified results by using traditional empirical approaches.
There exists a large variety of different types of optimization problems. Typically, we distinguish between at least the following main classes of problems for which the mathematical background is well understood, and for which numerical algorithms are available.

- Linear programming
- Quadratic programming
- Constrained nonlinear programming
- Dynamic programming
- Least squares, min-max, l_1-optimization
- Large scale optimization
- Semidefinite programming
- Non-smooth optimization
- Mixed-integer programming
- Optimal control
- Stochastic optimization
- Global optimization
- Multicriteria optimization

In this review, we consider only smooth, i.e., differentiable constrained nonlinear programming problems:

$$\begin{aligned} &\min f(x) && x \in \mathbb{R}^n \\ &g_j(x) = 0, && j = 1, ..., m_e \ , \\ &g_j(x) \geq 0, && j = m_e + 1, ..., m \ , \\ &x_l \leq x \leq x_u . \end{aligned} \tag{1}$$

Here, x is an n-dimensional parameter vector, also called the vector of design variables, and $f(x)$ is the objective function or cost function to be minimized under nonlinear equality and inequality constraints given by $g_j(x)$, $j = 1, \ldots, m$. It is assumed that these

functions are continuously differentiable in $\mathbb{R}^n$. The above formulation implies that we do not allow any discrete or integer variables. Besides this, we do not require any further mathematical structure of the model functions. For a discussion of non-smooth, optimization problems see *Non-Smooth Optimization*.

To facilitate the subsequent notation, we assume that upper and lower bounds x_u and x_l are not handled separately, i.e., that they are considered as general inequality constraints. Then we get the Nonlinear Programming Problem (NLP):

$$\begin{aligned} &\min f(x) && x \in \mathbb{R}^n \\ &g_j(x) = 0, && j = 1, ..., m_e \ , \\ &g_j(x) \geq 0, && j = m_e + 1, ..., m \ . \end{aligned} \tag{2}$$

Although optimization software can be used in the form of a *black box*, it is highly desirable to understand at least the basic ideas of the mathematical analysis behind the problem. One reason is that there are many situations, which could prevent an algorithm from approaching a solution in the correct way. Typically, an optimization algorithm breaks down with an error message and the corresponding documentation contains many technical phrases that must be understood in order to find a remedy. Another reason is that one would like to get an idea about how accurate the obtained solution is, and whether it is possible to improve or verify an existing approximation.

For these reasons, we present a very brief outline of the optimization theory behind the presented algorithms, on a very elementary level. First, we need some notations for the first and second derivatives of a differentiable function. For mathematical basics see *Differential Calculus*. The gradient of a real-valued function $f(x)$ is:

$$\nabla f(x) := \left(\frac{\partial}{\partial x_1} f(x), ..., \frac{\partial}{\partial x_n} f(x) \right)^T . \tag{3}$$

A second differentiation gives the Hessian matrix of $f(x)$.

$$\nabla^2 f(x) := \left(\frac{\partial^2}{\partial x_i \partial x_j} f(x) \right)_{i,\, j=1,...,n} . \tag{4}$$

The Jacobian matrix of a vector-valued function $F(x) = (f_1(x), \ldots, f_l(x))^T$ is

$$\nabla F(x) := \left(\frac{\partial}{\partial x_i} f_j(x) \right)_{i=1, ..., n;\, j=1, ..., l} , \tag{5}$$

also written in the form

$$\nabla F(x) := \left(\nabla f_1(x), ..., \nabla f_l(x)\right). \tag{6}$$

The fundamental tool for deriving optimality conditions and optimization algorithms is the so-called *Lagrangian function:*

$$L(x,u) := f(x) - \sum_{j=1}^{m} u_j g_j(x) \tag{7}$$

defined for all $x \in \mathbb{R}^n$ and $u = (u_1, ..., u_m)^T \in \mathbb{R}^m$. The purpose of $L(x, u)$ is to link the objective function $f(x)$ and constraints $g_j(x)$, $j = 1, \ldots, m$. The variables u_j are called the *Lagrangian multipliers* of the nonlinear programming problem.

Moreover, we denote the *feasible region* by P, i.e., the set of all feasible solutions:

$$P := \{x \in \mathbb{R}^n : g_j(x) = 0, j = 1, ..., m_e, g_j(x) \geq 0, j = m_e + 1, ..., m\}. \tag{8}$$

The *active inequality constraints* with respect to $x \in P$ are characterized by the index set:

$$I(x) := \{j : g_j(x) = 0, m_e < j \leq m\}. \tag{9}$$

We very briefly discuss the main strategies behind a few classes of nonlinear programming algorithms:

- Penalty and barrier methods
- Augmented lagrangian methods
- Interior point methods
- Sequential linear programming methods
- Sequential quadratic programming methods
- Generalized reduced gradient methods
- Sequential convex programming methods

In particular, we also discuss extensions of these methods to solve large-scale optimization problems.

Each implementation of a method in one of these subclasses requires additional decisions on a special variant or parameter selection, so that different codes of the same group may possess completely different performance characteristics in practice. Moreover, there exist combinations of the fundamental strategies, making it even more difficult to classify nonlinear programming algorithms. Comparative studies of codes for the general model have been performed in the past. They either proceed from randomly generated test examples or are based on artificial or simple application problems reflecting special mathematical properties.

2. Optimality Conditions

2.1 Convexity and Constraint Qualification

In general, we can only expect that an optimization algorithm computes a local minimum and not a global one; i.e., a point x^* with $f(x^*) \le f(x)$ for all $x \in P \cap U(x^*)$; where $U(x^*)$ is a suitable neighborhood of x. However, if the problem is convex, then each local minimum of a nonlinear programming problem is a global one. For example, if f is convex, g_j linear for $j = 1, \ldots, m_e$, and g_j concave for $j = m_e + 1, \ldots, m$, then these conditions force the feasible region P to be a convex set.

Definition 1

A function $f : \mathbb{R}^n \to \mathbb{R}$ *is called convex, if*

$$f(\lambda x + (1-\lambda) y) \le \lambda f(x) + (1-\lambda) f(y) \qquad (10)$$

for all $x, y \in \mathbb{R}^n$ *and* $\lambda \in (0,1)$, *and concave, if we replace* $' \le '$ *by* $' \ge '$ *in the above inequality.*

For a twice differentiable function f, convexity is equivalent to the property that $\nabla^2 f(x)$ is positive semidefinite; i.e., $z^T \nabla^2 f(x) z \ge 0$ for all $z \in \mathbb{R}^n$. Convexity of an optimization problem is important mainly from the theoretical point of view, since many convergence, duality or other theorems can be proved only for this special case. In practical situations, however, we have hardly a chance to test whether a numerical problem is convex or not.

To be able to formulate the subsequent optimality conditions, we need a special assumption, namely, to avoid irregular behavior of the feasible sets P at a local solution. We call it *constraint qualification*, which is to be considered as some kind of *regularity* in a more general form. In our situation, it is sufficient to proceed from the following definition.

Definition 2

A constraint qualification in $x^* \in P$ *is satisfied, if the gradients of active constraints, i.e., the vectors* $\nabla g_j(x^*)$ *for* $j \in \{1, ..., m_e\} \cup I(x^*)$, *are linearly independent.*

2.2 Karush-Kuhn-Tucker Conditions

For developing and understanding an optimization method, the subsequent theorems are essential. They characterize optimality and are, therefore, important for testing a current iterate with respect to its convergence accuracy.

Theorem 1 (Necessary Second Order Optimality Conditions)

Let f and g_j *be twice continuously differentiable for j = 1, …, m,* x^* *be a local minimizer of (2) and the constraint qualification in* x^* *be satisfied. Then there exists a* $u^* \in \mathbb{R}^m$, *such that:*

$$
\begin{aligned}
\text{a)} \qquad u_j^* &\geq 0\,, \quad j = m_e + 1, \ldots, m\,,\\
g_j(x^*) &= 0\,, \quad j = 1, \ldots, m_e\,,\\
g_j(x^*) &\geq 0\,, \quad j = m_e + 1, \ldots, m\,,\\
\nabla_x L(x^*, u^*) &= 0\,,\\
u_j^* g_j(x^*) &= 0\,, \quad j = m_e + 1, \ldots, m\,,\\
&(\textit{first order condition}),
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
&b)\ \ s^T \nabla_x^2 L(x^*, u^*) s \geq 0\\
&\quad \textit{for all } s \in \mathbb{R}^n \textit{ with } \nabla g_j(x^*)^T\\
&\quad s = 0,\ j \in \{1, \ldots, m_e\} \cup I(x^*)\ (\textit{second order condition}).
\end{aligned}
\tag{12}
$$

Statement a) of the theorem is called the *Karush-Kuhn-Tucker condition*. It says that at a local solution, the gradient of the objective function can be expressed by a linear combination of gradients of active constraints. Moreover, statement b) implies that the Lagrangian function is positive semidefinite on the tangential space defined by the active constraints. For a discussion of general duality-based optimality conditions see *Duality Theory*.

It is not possible to omit the constraint qualification, as shown by the subsequent example.

Example 1

Let

$$
\begin{aligned}
&f(x_1, x_2) := x_1,\\
&g_1(x_1, x_2) := -x_2 \geq 0,\\
&g_2(x_1, x_2) := x_2 - x_1^2 \geq 0.
\end{aligned}
\tag{13}
$$

Since P = {(0, 0)}, $x^* = (0, 0)$ is the optimal solution. However, we have

$$
\nabla_x L(x^*, u^*) = \begin{pmatrix} 1 \\ u_1^* - u_2^* \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\tag{14}
$$

showing that the Karush-Kuhn-Tucker-condition cannot be satisfied.

It is also possible to derive a very similar reverse optimality condition that does not require the constraint qualification.

Theorem 2 (Sufficient Second Order Optimality Conditions)

Let f and g_j be twice continuously differentiable for j = 1, …, m and $x^* \in \mathbb{R}^n$, $u^* \in \mathbb{R}^m$ be given, so that the following conditions are satisfied

$$
\begin{aligned}
\text{a)} \qquad u_j^* &\geq 0, \quad j = m_e + 1, \ldots, m, \\
g_j(x^*) &= 0, \quad j = 1, \ldots, m_e, \\
g_j(x^*) &\geq 0, \quad j = m_e + 1, \ldots, m, \\
\nabla_x L(x^*, u^*) &= 0, \\
u_j^* g_j(x^*) &= 0, \quad j = m_e + 1, \ldots, m, \\
&(\textit{first order condition}),
\end{aligned}
\tag{15}
$$

$$
\begin{aligned}
&b)\ s^T \nabla_x^2 L(x^*, u^*) s > 0 \\
&\textit{for all } s \in \mathbb{R}^n \textit{ with } s \neq 0, \nabla g_j(x^*)^T s = 0,\ j = 1, \ldots, m_e, \textit{ and for all } s \textit{ with} \\
&\textit{with } \nabla g_j(x^*)^T s = 0,\ j = m_e + 1, \ldots, m, \textit{ and } u_j^* > 0 \\
&(\textit{second order condition}).
\end{aligned}
\tag{16}
$$

Then x^* is an isolated local minimum of f on P, i.e., there is a neighborhood $U(x^*)$ of x^* with $f(x^*) < f(x)$ for all $x \in U(x^*) \cap P, x \neq x^*$.

When reading a nonlinear programming textbook, one has to be aware of the fact that the optimality conditions are often stated in a slightly different way. The formulation of a NLP problem varies from author to author, depending upon whether a minimum or a maximum is searched, whether the inequality constraints use ≤ instead of ≥, or whether upper and lower bounds are included. Also, there exist different versions of the above statements, where the assumptions are either more general or more specialized.

Now let us consider a few examples.

Example 2

Assume that $n = 2$, $m_e = 0$, $m = 2$, and that x is an optimal solution with active constraints g_1 and g_2. Then the gradient of the objective function must point into the cone spanned by the gradients $\nabla g_1(x^*)$ and $\nabla g_2(x^*)$. In other words, there must exist two multipliers $u_1^* \geq 0$ and $u_2^* \geq 0$ with

$$\nabla f(x^*) = u_1^* \nabla g_1(x^*) + u_2^* \nabla g_2(x^*). \tag{17}$$

Example 3

Consider the simple NLP

$$
\begin{aligned}
f(x) &:= x_1^2 + x_2, \\
g_1(x) &:= 9 - x_1^2 - x_2^2 \geq 0, \\
g_2(x) &:= 1 - x_1 - x_2 \geq 0.
\end{aligned}
\tag{18}
$$

We observe immediately that $x^* = (0, -3)^T$ is the unique optimal solution of the convex

optimization problem. From the Karush-Kuhn-Tucker condition

$$\nabla_x L(x,u) = \begin{pmatrix} 2x_1 \\ 1 \end{pmatrix} - u_1 \begin{pmatrix} -2x_1 \\ -2x_2 \end{pmatrix} - u_2 \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 2x_1(1+u_1)+u_2 \\ 1+2u_1x_2+u_2 \end{pmatrix} = 0 \qquad (19)$$

we get the multipliers $u_1^* = 1/6$ *and* $u_2^* = 0$. *Moreover, the Hessian matrix of the Lagrangian function*

$$\nabla_x^2 L(x^*,u^*) = \begin{pmatrix} 7/3 & 0 \\ 0 & 1/3 \end{pmatrix} \qquad (20)$$

is positive definite.

-
-
-

Bibliography

Biegler L.T., Coleman T.F., Conn A.R., and Santosa F.N. eds. (1997). *Large-Scale Optimization with Applications, Part I, II, III*, pp 105. Heidelberg, New York: Springer. [A series on papers about large-scale optimization and applications]

Coleman T.F. (1984). *Large Sparse Numerical Optimization*. Lecture Notes in Computer Science **165**. Heidelberg, New York: Springer-Verlag. [This is an overview of basic techniques for large scale nonlinear programming]

Edgar T.F. and Himmelblau D.M. (1988). *Optimization of Chemical Processes*, pp. 652. New York: McGraw Hill. [A book about optimization models and algorithms, especially for chemical engineering]

Fiacco A.V. and McCormick G.P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, pp. 210. New York: Wiley. Reprinted by Philadelphia: SIAM Publications (1990). [This book is a classical introduction to penalty and barrier methods]

Fletcher R. (1987). *Practical Methods of Optimization*, pp. 436. New York: John Wiley and Sons. [This is an introduction to theory and classical methods]

Gill P.E., Murray W., and Wright M.H. (1981, 1992). *Practical Optimization*, pp. 401. New York: Academic Press. [An overview of optimization methods from a numerical point of view]

Kaplan W. (1999). *Maxima and Minima with Applications: Practical Optimization and Duality*, pp. 284. New York: John Wiley [Introduction into more theoretical aspects of nonlinear programming]

Luenberger D.G. (1984). *Linear and Nonlinear Programming*, pp. 491. Reading, Massachusetts: Addison-Wesley. [Introduction to theory and classical methods]

Mangasarian O.L. (1969). *Nonlinear Programming*, pp. 220 .New York: McGraw-Hill. Reprinted by Philadelphia: SIAM Publications (1994). [An overview of the theoretical background of non-linear programming, esp. optimality conditions]

Moré J.J. and Wright S.J. (1993). *Optimization Software Guide*, pp. 154. Philadelphia: SIAM Publications. [This is a review on available nonlinear programming software]

Nesterov Y.E. and Nemirovskii A.S. (1994). *Interior Point Polynomial Methods in Convex Programming*. Philadelphia: SIAM Publications. [A comprehensive study of the theoretical background of interior point methods]

Nocedal J. and Wright S.J. (1999). *Numerical Optimization*, pp. 636. Heidelberg, New York: Springer. [A more modern introduction and review on nonlinear programming techniques]

Papalambros P. and Wilde D.J. (1988,2000). *Principles of Optimal Design: Modeling and Computation*, pp. 416. Cambridge: Cambridge University Press. [An introduction with applications from mechanical engineering]

Schittkowski K. (1980) *Nonlinear Programming Codes - Information, Tests, Performance*, pp.242. Heidelberg, New York: Springer, Lecture Notes in Economics and Mathematical Systems, Vol. 183 [Comparative performance results for a large number of nonlinear programming codes]

Schittkowski K. (2002). *Numerical Data Fitting in Dynamical Systems*, pp. 403. Dordrecht, Boston, London: Kluwer Academic Publishers [Least squares algorithms, numerical experiments, and software]

Wright S.J. (1997). *Primal-Dual Interior-Point Methods*, pp. 289. Philadelphia: SIAM publications. [An introduction to interior point methods for linear programming with extensions to nonlinear programming]

Biographical Sketches

Klaus Schittkowski is Professor for Applied Computer Science at the Department of Mathematics of the University of Bayreuth, Germany. He received his diploma and doctoral degrees at the University of Würzburg, Department of Applied Mathematics. His main research areas are mathematical optimization, nonlinear programming, numerical algorithms, and applications. He is the author of about 65 scientific publications, among them five monographs. Moreover he implemented a couple of numerical algorithms, for nonlinear programming, data fitting, parameter estimation in dynamical systems, and optimal control, that are used worldwide, in academia and industry. He has worked on numerous cooperative projects with firms, for example, AEDS Airbus, BASF, Bayer, Boehringer Ingelheim, Siemens, and Epcos. To disseminate knowledge about mathematical modeling and optimization, he organized more than 10 tutorials, especially in structural design optimization.

Christian Zillober is Research Assistant at the Department of Mathematics of the University of Bayreuth, Germany. He received his diploma and doctoral degree at the Technical University of Munich, Department of Applied Mathematics and Statistics. His research area is nonlinear programming, especially related to mechanical engineering. He implemented algorithms for numerical optimization, which are used in several academic departments and in industry.